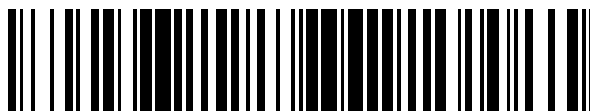


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 741 998**

51 Int. Cl.:

G06F 9/30 (2008.01)

G06F 9/38 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **18.08.2010 PCT/US2010/045910**

87 Fecha y número de publicación internacional: **24.02.2011 WO11022493**

96 Fecha de presentación y número de la solicitud europea: **18.08.2010 E 10750200 (7)**

97 Fecha y número de publicación de la concesión europea: **15.05.2019 EP 2467776**

54 Título: **Procedimientos y aparatos para predecir la no ejecución de instrucciones de no bifurcación condicional**

30 Prioridad:

19.08.2009 US 543847

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

12.02.2020

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)
International IP Administration 5775 Morehouse
Drive
San Diego, CA 92121, US**

72 Inventor/es:

**STEMPEL, BRIAN M.;
DIEFFENDERFER, JAMES N.;
SARTORIUS, THOMAS A.;
MANDZAK, DAVID J. y
SMITH, RODNEY W.**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 741 998 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Procedimientos y aparatos para predecir la no ejecución de instrucciones de no bifurcación condicional

5 **Campo de la divulgación**

[0001] La presente divulgación se refiere en general al campo de los procesadores y, en particular, a los procesadores que admiten instrucciones de no bifurcación condicional.

10 **Antecedentes**

[0002] Muchos productos portátiles, tales como teléfonos móviles, ordenadores portátiles, asistentes de datos personales (PDA) y similares, utilizan un sistema de procesamiento que ejecuta programas, tales como programas de comunicación y multimedia. Un sistema de procesamiento para dichos productos puede incluir múltiples procesadores, sistemas de memoria complejos para almacenar instrucciones y datos, controladores, dispositivos periféricos tales como interfaces de comunicación y bloques lógicos de función fija configurados, por ejemplo, en un único chip. Al mismo tiempo, los productos portátiles tienen una fuente de energía limitada en forma de baterías a las que a menudo se requiere que presten soporte a operaciones de alto rendimiento por parte del sistema de procesamiento. Para aumentar la duración de la batería, se desea realizar estas operaciones de la manera más eficaz posible. Muchos ordenadores personales también se están desarrollando con diseños eficaces para funcionar con un consumo de energía total reducido.

[0003] Los procesadores emplean una arquitectura de procesamiento en cadena (pipeline) con un conjunto de instrucciones que incluye, en general, instrucciones de bifurcación condicional. Los programas pueden usar las instrucciones de bifurcación condicional para controlar el flujo de las operaciones de programa. Sin embargo, la ejecución de instrucciones de bifurcación condicional puede causar una burbuja en el procesamiento en cadena pendiente de la resolución de la condición de bifurcación asociada, que, en general, no se determina hasta una fase profunda en el procesamiento en cadena del procesador. Muchos procesadores también incluyen instrucciones de no bifurcación condicional para ayudar a aliviar las propiedades de reducción de rendimiento de las instrucciones de bifurcación condicional. La ejecución condicional de instrucciones de no bifurcación permite que un programador especifique si una instrucción debe ejecutarse o no según un estado de máquina generado previamente. El uso de instrucciones de no bifurcación condicional ayuda a reducir la necesidad de instrucciones de bifurcación condicional y, por lo tanto, mejora el rendimiento. El documento WO 00/41071 A1 describe un procedimiento y un aparato para determinar un valor de predicado predicho para un predicado. Una instrucción de predicado se ejecuta después de manera condicional dependiendo del valor de predicado predicho. El documento US 6442679 B1 describe un aparato de guardia para predecir resultados de guardia para instrucciones de predicado, cada una de las cuales especifica un operador de guardia que se aplicará a una fuente de guardia para generar el resultado de guardia.

[0004] Cuando se evalúa una condición asociada de una instrucción condicional e indica que no se debe ejecutar la instrucción, es posible que los recursos asociados a la instrucción condicional ya se hayan consumido. Por ejemplo, los operandos de registro requeridos para que se ejecute la instrucción de no bifurcación condicional ya pueden haberse recuperado. Además, la instrucción de no bifurcación condicional puede haber introducido innecesariamente dependencias de procesamiento en cadena en el procesamiento en cadena del procesador. Por ejemplo, una instrucción condicional puede detenerse en el procesamiento en cadena mientras espera a que se resuelva su condición, lo que provoca que la detención se extienda a todas las instrucciones que dependen de la ejecución de la instrucción condicional. Además, en un bucle de software puede haber instrucciones condicionales, con sus propiedades de resolución de condiciones que se producen de manera similar en cada iteración del bucle, lo que puede causar una degradación significativa del rendimiento.

50 **SUMARIO**

[0005] Entre sus diversos aspectos, la presente divulgación reconoce que proporcionar procedimientos y aparatos más eficaces para predecir la no ejecución de instrucciones de no bifurcación condicional puede mejorar el rendimiento y reducir los requisitos de potencia en un sistema de procesador. Para dichos fines, un modo de realización de la invención se refiere a un procedimiento para no ejecutar una instrucción de no bifurcación condicional emitida. Una instrucción de no bifurcación condicional se identifica como idónea para una predicción, donde la predicción indica que la instrucción de no bifurcación condicional idónea (ECNB) no va a ejecutarse. La instrucción ECNB se ejecuta como una instrucción de no operación (NOP) en respuesta a la predicción de que la instrucción ECNB no va a ejecutarse.

[0006] Otro modo de realización se refiere un aparato para predecir que una instrucción de no bifurcación condicional no va a ejecutarse. El aparato tiene un primer circuito para identificar una instrucción de no bifurcación condicional como idónea para una predicción. El aparato tiene un segundo circuito para predecir si la instrucción de no bifurcación condicional idónea (ECNB) no se ejecutará en respuesta al cumplimiento de un criterio de evaluación.

5 **[0007]** Otro modo de realización se refiere un procedimiento para predecir que una instrucción de no bifurcación condicional no va a ejecutarse. Se identifica una instrucción de no bifurcación condicional que es idónea para predecir si se ejecutará o no. Se predice que la instrucción de no bifurcación condicional idónea (ECNB) no se ejecutará en respuesta al cumplimiento de un criterio de evaluación.

10 **[0008]** Se entiende que otros modos de realización de la presente invención resultarán fácilmente evidentes para los expertos en la técnica a partir de la siguiente descripción detallada, en la que se muestran y se describen a modo de ilustración diversos modos de realización de la invención. Como se comprobará, la invención es susceptible de otros modos de realización diferentes y sus diversos detalles pueden modificarse en otros diversos aspectos, todo sin apartarse del alcance de la presente invención. En consecuencia, los dibujos y la descripción detallada han de considerarse de naturaleza ilustrativa y no restrictiva.

15 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

[0009] Diversos aspectos de la presente invención se ilustran a modo de ejemplo, y no de manera limitativa, en los dibujos adjuntos, en los que:

20 La FIG. 1 ilustra un sistema de comunicación inalámbrica;

la FIG. 2 muestra un sistema de procesador ejemplar que predice si ejecutar o no ejecutar instrucciones de no bifurcación condicional;

25 la FIG. 3 ilustra un circuito de predicción de instrucción de no bifurcación condicional idónea (ECNB) ejemplar;

la FIG. 4A ilustra un primer proceso para predecir la ejecución de una instrucción ECNB;

la FIG. 4B ilustra un segundo proceso para predecir la ejecución de una instrucción ECNB;

30 la FIG. 5 ilustra un tercer proceso para predecir la ejecución de una instrucción ECNB; y

la FIG. 6 ilustra un cuarto proceso para predecir la ejecución de una instrucción ECNB.

35 **DESCRIPCIÓN DETALLADA**

[0010] La descripción detallada expuesta a continuación en relación con los dibujos adjuntos está concebida como una descripción de diversos modos de realización a modo de ejemplo de la presente invención y no está concebida para representar los únicos modos de realización en los que puede llevarse a la práctica la presente invención. La descripción detallada incluye detalles específicos con el propósito de proporcionar un entendimiento exhaustivo de la presente invención. Sin embargo, a los expertos en la técnica les resultará evidente que la presente invención puede llevarse a la práctica sin estos detalles específicos. En algunos casos, estructuras y componentes bien conocidos se muestran en forma de diagrama de bloques para no oscurecer los conceptos de la presente invención.

45 **[0011]** La FIG. 1 ilustra un sistema de comunicación inalámbrica 100 a modo de ejemplo en el que se puede emplear de forma ventajosa un modo de realización de la invención. Con propósitos ilustrativos, la FIG. 1 muestra tres unidades remotas 120, 130 y 150 y dos estaciones base 140. Se reconocerá que los típicos sistemas de comunicación inalámbrica pueden tener muchas más unidades remotas y estaciones base. Las unidades remotas 120, 130, 150 y las estaciones base 140 que incluyen componentes de hardware, componentes de software, o ambos, como se representa mediante los componentes 125A, 125C, 125B y 125D, respectivamente, se han adaptado para realizar la invención como se analiza más adelante a continuación. La FIG. 1 muestra señales de enlace directo 180 desde las estaciones base 140 a las unidades remotas 120, 130 y 150, y señales de enlace inverso 190 desde las unidades remotas 120, 130 y 150 a las estaciones base 140.

55 **[0012]** En la FIG. 1, la unidad remota 120 se muestra como un teléfono móvil, la unidad remota 130 se muestra como un ordenador portátil y la unidad remota 150 se muestra como una unidad remota de ubicación fija en un sistema de bucle local inalámbrico. A modo de ejemplo, las unidades remotas pueden ser, de forma alternativa, teléfonos móviles, radiolocalizadores, walkie talkies, unidades manuales de sistemas de comunicación personal (PCS), unidades de datos portátiles, tales como asistentes de datos personales, o unidades de datos de ubicación fija, tales como equipos de lectura de contadores. Aunque la FIG. 1 ilustra unidades remotas de acuerdo con las enseñanzas de la divulgación, la divulgación no está limitada a estas unidades ejemplares ilustradas. Los modos de realización de la invención pueden emplearse de forma adecuada en un procesador que tiene instrucciones de no bifurcación condicional.

65 **[0013]** La FIG. 2 muestra un sistema de procesador 200 ejemplar que predice si ejecutar o no ejecutar instrucciones de no bifurcación condicional. El sistema de procesador 200 incluye un procesador 210, un sistema

de memoria caché 212, una memoria de sistema 214 y un sistema de entrada y salida (E/S) 216. El procesador 210 comprende, por ejemplo, un procesamiento en cadena de instrucciones 220 y un circuito lógico de predicción de no bifurcación condicional 222. El sistema de memoria caché 212, por ejemplo, comprende una memoria caché de instrucciones (Icaché) 224, un controlador de memoria 226 y una memoria caché de datos (Dcaché) 228. La memoria de sistema 214 proporciona acceso a las instrucciones y datos que no se encuentran en Icaché 224 o en la Dcaché 228. Cabe señalar que el sistema de memoria caché 212 se puede integrar con el procesador 210 y también puede incluir múltiples niveles de memoria caché en una organización jerárquica. El sistema de E/S 216 comprende una pluralidad de dispositivos de E/S, tales como dispositivos de E/S 240 y 242, que interactúan con el procesador 210.

[0014] El procesamiento en cadena de instrucciones 220 está formada por una serie de etapas, tales como, una etapa de extracción 230, una etapa de descodificación 231, una etapa de emisión 232, una etapa de ejecución 233 y una etapa de finalización 234. Los expertos en la materia reconocerán que cada etapa 230 a 234 en el procesamiento en cadena de instrucciones 220 puede comprender una serie de etapas adicionales de procesamiento en cadena, por ejemplo, en función de la frecuencia operativa del procesador y de la complejidad de las operaciones requeridas en cada etapa. Además, la etapa de ejecución puede estar constituida por uno o más circuitos de etapa de ejecución de instrucciones, tales como un sumador, un multiplicador, operaciones lógicas, operaciones de desplazamiento y rotación, y similares. Dichos circuitos de etapa de ejecución de instrucciones pueden estar asociados a instrucciones de no bifurcación condicional. Cada una de las etapas de procesamiento en cadena puede tener implementaciones variadas sin apartarse de los procedimientos y aparatos de predicción condicional descritos en el presente documento.

[0015] La etapa de extracción 230 obtiene instrucciones para su ejecución desde la memoria caché de instrucciones (Icaché) 224 de acuerdo con un flujo de programa informático que puede incluir instrucciones de bifurcación condicional e instrucciones de no bifurcación condicional. En general, una instrucción de bifurcación condicional extraída usa lógica de predicción de bifurcación para predecir si se tomará la bifurcación condicional. Una instrucción de no bifurcación extraída que no es una instrucción de no bifurcación condicional avanza hasta la etapa de descodificación 231 para ser descodificada, emitida para su ejecución en la etapa de emisión 232, ejecutada en la etapa de ejecución 233 y retirada en la etapa de finalización 234. Una instrucción de no bifurcación condicional extraída utiliza el circuito lógico de predicción de no bifurcación condicional 222 como se describe en el presente documento para determinar si la instrucción no debe ejecutarse. Una instrucción de no bifurcación condicional que no se ejecuta no cambia el estado de procesador tal como existía antes de encontrar la instrucción de no bifurcación condicional.

[0016] El circuito lógico de predicción de no bifurcación condicional 222 comprende un circuito lógico de detección 246, un circuito lógico de supervisión 248 que tiene un filtro 250 y una tabla de historial condicional 252 y un circuito lógico de predicción y corrección 254. En un modo de realización, se supone que una mayoría de las instrucciones de no bifurcación condicional tienen, en general, sus condiciones resueltas en el mismo valor para la mayoría de las iteraciones de un bucle de software.

[0017] El circuito lógico de detección 246, como un detector de bucle de software, funciona basándose en las características dinámicas de instrucciones de bifurcación condicional usadas en bucles de software. En bucles de software con una sola entrada y una sola salida, una bifurcación de fin de bucle es, en general, una instrucción de bifurcación condicional que se bifurca hacia el inicio del bucle de software para todas las iteraciones del bucle, a excepción de la última iteración, que sale del bucle de software. El circuito lógico de detección 246 puede tener múltiples modos de realización para la detección de bucles de software como se describe con más detalle a continuación y en la solicitud de patente estadounidense 11/066.508, cedida al cesionario de la presente solicitud, titulada "Suppressing Update of a Branch History Register by Loop-Ending Branches [Supresión de actualización de un registro de historial de bifurcación por bifurcaciones de fin de bucle]".

[0018] De acuerdo con un ejemplo, se supone que cada instrucción de bifurcación condicional con una dirección de destino de bifurcación menor que la dirección de instrucción de bifurcación condicional, y por lo tanto considerada una bifurcación hacia atrás, es una instrucción de bifurcación de fin de bucle. Esto requiere una comparación de direcciones cuando se determina la dirección de destino de bifurcación. Dado que no todas las bifurcaciones hacia atrás son bifurcaciones de fin de bucle, existe un cierto nivel de imprecisión que debería tenerse en cuenta.

[0019] En otro ejemplo, una bifurcación de fin de bucle puede ser detectada en bucles simples mediante el reconocimiento de la ejecución repetida de la misma instrucción de bifurcación. Al almacenar el valor del contador de programa para la última instrucción de bifurcación hacia atrás en un registro de propósito especial, y comparar este valor almacenado con la dirección de instrucción de la siguiente instrucción de bifurcación hacia atrás, se puede reconocer una bifurcación de fin de bucle cuando las dos direcciones de instrucción coinciden. Como el código puede incluir instrucciones de bifurcación condicional dentro de un bucle de software, la determinación de la instrucción de bifurcación de fin de bucle puede volverse más complicada. En dicha situación, múltiples registros de propósito especial pueden ser instanciados en hardware para almacenar las direcciones de instrucciones de

cada instrucción de bifurcación condicional. Al comparar con todos los valores almacenados, se puede determinar una coincidencia para la bifurcación de fin de bucle.

5 **[0020]** Además, una bifurcación de fin de bucle puede estar marcada estáticamente por un compilador o ensamblador. Por ejemplo, un compilador genera un tipo particular de instrucción de bifurcación, mediante el uso de un código de operación único, o mediante la configuración de un campo de bits de formato especial, que solo se usa para bifurcaciones de fin de bucle. Al descodificar la instrucción de bifurcación particular, se determina la bifurcación de fin de bucle.

10 **[0021]** El circuito lógico de supervisión 248 comprende un filtro 250, una tabla de historial condicional (CHT) 252 y la lógica de supervisión asociada. En un modo de realización, un proceso de supervisión guarda información de estado de eventos de condición preestablecidos que pueden haberse producido en una o más ejecuciones previas de un bucle de software que tiene una instrucción de no bifurcación condicional que es idónea para predicción. En un modo de realización, todas las instrucciones de no bifurcación condicional pueden no ser idóneas para la predicción. Por ejemplo, las instrucciones de no bifurcación condicional implementadas con microcódigo, por razones de complejidad de implementación, pueden no ser idóneas para una operación de ejecución prevista. Además, las instrucciones de bifurcación condicional no serían idóneas para la predicción de instrucciones de no bifurcación condicional, ya que las instrucciones de bifurcación tienen, en general, su propio hardware y procedimientos de predicción que funcionan de manera diferente a las técnicas de predicción descritas en el presente documento.

25 **[0022]** La información histórica se usa para predecir cuándo no se ejecutará una instrucción de no bifurcación condicional idónea (ECNB). Como se describe con más detalle a continuación, los enfoques se usan para determinar con alta confianza si una instrucción ECNB se ejecutará o no. Los enfoques para determinar procedimientos de predicción de alta confianza son ventajosos, ya que la penalización por predecir que una instrucción ECNB no se ejecute cuando debería ejecutarse es más grave que predecir que una instrucción ECNB se ejecute cuando no debería ejecutarse. Por ejemplo, una instrucción ECNB que se predice que no se ejecutará cambiará las operaciones de procesamiento en cadena asociadas a la instrucción ECNB para minimizar la potencia y/o mejorar el rendimiento al no realizar operaciones ECNB seleccionadas que no serían necesarias cuando se predice que la instrucción ECNB no se ejecutará. Por ejemplo, un operando de memoria especificado por una instrucción de carga condicional no tendría que extraerse si se predice que la instrucción de carga condicional no va a ejecutarse. Para una instrucción ECNB de este tipo que se predice que no se ejecutará, el procesamiento en cadena se cambiaría en la etapa de procesamiento en cadena adecuada, por ejemplo, para no extraer ningún registro u operandos de memoria necesarios para la ejecución de la instrucción, a fin de reducir la potencia y mejorar el rendimiento. Sin embargo, si la condición especificada por la instrucción ECNB predicha indica una predicción incorrecta, el procesamiento en cadena debe inicializarse al menos en el punto del código extraído donde se pueden corregir los efectos debidos a la predicción incorrecta. Una instrucción ECNB que se predice que se ejecutará cuando no debería ejecutarse no requiere una inicialización de procesamiento en cadena, sino que, en el caso de una predicción incorrecta, finaliza la instrucción de tal manera que el estado del procesador no se vea afectado.

45 **[0023]** Un proceso de evaluación de condición evalúa la información de estado guardada de los eventos de condición preestablecidos y tras cumplir con un criterio de evaluación preestablecido, permite la predicción de una instrucción de no bifurcación condicional idónea (ECNB) presente para su próxima ejecución en el bucle. Por ejemplo, un evento de condición preestablecido puede incluir un número preestablecido de veces que se debe ejecutar un bucle de software y si una o más instrucciones ECNB anteriores se ejecutaron o no se ejecutaron en función del estado de la condición asociada. Por ejemplo, los criterios de evaluación preestablecidos pueden incluir cumplir con un número establecido de iteraciones de un bucle de software y tener un estado previo de no ejecutar una instrucción ECNB anterior encontrada en el número establecido anterior de iteraciones de bucle. Por ejemplo, el criterio de evaluación preestablecido puede requerir que no se ejecuten las instrucciones ECNB anteriores encontradas en dos ejecuciones previas del bucle de software. En tal caso, se predice que la presente instrucción ECNB no se ejecutará en la siguiente iteración del bucle de software.

55 **[0024]** Para dar soporte a un circuito lógico de supervisión 248 de este tipo, el filtro 250 determina si una instrucción de no bifurcación condicional extraída es idónea para la ejecución predicha. Si una instrucción extraída no es idónea para la ejecución predicha, la instrucción extraída se ejecuta según lo especificado por la arquitectura del procesador sin la ayuda de información de predicción. Si una instrucción extraída es idónea para una ejecución predicha, la CHT 252 se habilita. Se selecciona una entrada de la CHT 252, asociada con una instrucción ECNB, para proporcionar información de predicción a la lógica de predicción que forma parte del circuito lógico de predicción y corrección 254. Dicha información de predicción es rastreada, por ejemplo, por las etapas de procesamiento en cadena 232-234 a medida que la instrucción ECNB se mueve a través del procesamiento en cadena.

65 **[0025]** La entrada de CHT 252 registra la historia de la ejecución de la instrucción extraída idónea para su ejecución predicha. Por ejemplo, cada entrada de la CHT puede comprender una combinación de valores de conteo de contadores de estado de ejecución y de bits de estado que son entradas para la lógica de predicción. La CHT

252 también puede comprender una lógica de índice para permitir que una instrucción ECNB extraída sirva como índice para una entrada en la CHT 252 asociada a la instrucción ECNB extraída, ya que pueden existir múltiples instrucciones ECNB en un bucle de software. Por ejemplo, al contar el número de instrucciones ECNB desde el extremo superior de un bucle de software, el conteo puede usarse como un índice para la CHT 252. El circuito lógico de supervisión 248 incluye contadores de bucle para contar iteraciones de bucles de software y garantizar que los contadores de estado de ejecución hayan tenido la oportunidad de saturarse en un valor de conteo especificado que represente, por ejemplo, un estado fuertemente no ejecutado. Si un contador de estado de ejecución se ha saturado, la lógica de predicción se habilita para predecir la no ejecución de la instrucción asociada de no bifurcación condicional extraída en la siguiente iteración del bucle.

[0026] La lógica de predicción y corrección 254 genera información de predicción que se rastrea en la etapa de emisión 232, la etapa de ejecución 233 y la etapa de terminación 234 en la emisión de registro de rastreo (TrI) 262, la ejecución de registro de rastreo (TrE) 263 y la completitud de registro de rastreo (TrC) 264. Por ejemplo, al predecir la no ejecución de la instrucción ECNB, la instrucción ECNB se trata de manera efectiva, por ejemplo, como una instrucción de no operación (NOP) en las etapas de procesamiento en cadena 232-234. Al tratar la instrucción ECNB como una NOP, los registros de propósito general (GPR), si son necesarios cuando se ejecuta una instrucción ECNB, no se leen, ya que no son necesarios para ejecutar una instrucción NOP predicha. Si la instrucción ECNB era una instrucción de acceso a memoria de carga o almacenamiento, la operación de acceso a memoria no se inicia como una instrucción NOP predicha. Por ejemplo, un circuito de extracción de operandos 235 que funciona en la etapa de ejecución 233 no extraerá un operando requerido para que la instrucción ECNB se ejecute en respuesta a una predicción de no ejecución. Al no leerse los GPR o acceder a la memoria, la potencia puede reducirse en el procesador 210. Además, el rendimiento del procesador puede mejorarse al no leer los GPR o al acceder a memoria y esperar innecesariamente operandos que no serían necesarios cuando la instrucción ECNB se predice como una NOP.

[0027] Al llegar a la etapa de ejecución 233, si la condición de ejecución especificada para la instrucción ECNB se ha evaluado de manera opuesta a su predicción, se corrige la ejecución de procesamiento en cadena de la instrucción NOP predicha. Por ejemplo, una corrección en el procesamiento en cadena puede incluir inicializar las instrucciones en cadena que comiencen en la etapa en la que se realizó la predicción. En un modo de realización alternativo, el procesamiento en cadena puede inicializarse desde la etapa de extracción inicial, donde la instrucción ECNB se extrajo inicialmente. Además, la entrada de CHT adecuada también puede corregirse después de una predicción incorrecta.

[0028] la FIG. 3 ilustra un circuito de predicción de instrucción de no bifurcación condicional idónea (ECNB) 300 ejemplar. El circuito de predicción de ECNB 300 ilustra circuitos y trayectorias de señales de control entre circuitos. Con más detalle, el circuito de predicción de instrucciones ECNB 300 incluye un circuito de detección 304, un circuito de supervisión 306 y un circuito de predicción y corrección 308. El circuito de supervisión 306 comprende un circuito de filtro 310 y un circuito de tabla de historial condicional (CHT) 312. El circuito de predicción y corrección 308 comprende un circuito de predicción 314, un circuito de rastreo 316 y un circuito de corrección 318.

[0029] El circuito de detección 304, que actúa como un detector de bucle, funciona para detectar una bifurcación de fin de bucle como se analizó anteriormente con respecto al circuito lógico de detección 246. Por ejemplo, una bifurcación de final de bucle es, en general, una instrucción de bifurcación condicional que se bifurca hacia el inicio del bucle para todas las iteraciones del bucle, excepto para la última iteración que sale del bucle. La información relativa a cada bucle identificado se pasa al circuito de filtro 310.

[0030] En un modo de realización, el circuito de filtro, por ejemplo, es un contador de bucle que proporciona una indicación de que se ha producido un número determinado de iteraciones de un bucle de software, tal como tres iteraciones de un bucle particular. Para cada iteración del bucle, el filtro determina si una instrucción de no bifurcación condicional es idónea para la predicción. Si una instrucción de no bifurcación condicional idónea (ECNB) está en el bucle, el estado de ejecución de la instrucción ECNB se registra en el circuito de la tabla de historial condicional (CHT) 312. Por ejemplo, un contador de estado de ejecución se puede usar para registrar un historial de ejecución de intentos de ejecución previos de una instrucción ECNB. Un contador de estado de ejecución puede actualizarse en una dirección para indicar una instrucción ECNB ejecutada de forma condicional y en un sentido opuesto para indicar que una instrucción ECNB no se ejecutó de forma condicional. Por ejemplo, se puede usar un contador de estado de ejecución de dos bits donde un estado no ejecutado causa una disminución del contador y un estado ejecutado causa un incremento del contador. A los estados de salida del contador de estado de ejecución se asignan, por ejemplo, una salida de "11" para indicar que se indica fuertemente que las instrucciones ECNB anteriores han sido ejecutadas, una salida de "10" para indicar que se indica de forma incierta que las instrucciones ECNB anteriores han sido ejecutadas, una salida de "01" para indicar que se indica de forma incierta que las instrucciones ECNB anteriores no han sido ejecutadas, y una salida de "00" para indicar que se indica fuertemente que las instrucciones ECNB anteriores no han sido ejecutadas. La salida "11" y la salida "00" de contador de estado de ejecución serán valores de salida saturados. Un contador de estado de ejecución se asociará a, o proporcionará estado para, cada instrucción ECNB en un bucle de software detectado. Sin embargo, una implementación particular puede limitar el número de contadores de estado de ejecución que se usan en la implementación y, por lo tanto, limitar el número de instrucciones ECNB que pueden predecirse. El

circuito de detección 304 reajusta, en general, los contadores de estado de ejecución tras la primera entrada a un bucle de software.

5 **[0031]** De forma alternativa, un indicador de inhabilitación de predicción puede estar asociado a cada instrucción ECNB a predecir, en lugar de a un contador de estado de ejecución. El indicador de inhabilitación de predicción se activa para inhabilitar la predicción si se ha determinado previamente que se ha ejecutado una instrucción ECNB asociada. Disponer de una instrucción ECNB previa que se ha ejecutado implica que el nivel de confianza para predecir una situación de no ejecución para la instrucción ECNB sería inferior a lo que puede ser aceptable.

10 **[0032]** Un contador de índice también se puede usar con la CHT 312 para determinar qué instrucción ECNB se está contando o evaluando en el bucle de software. Por ejemplo, en un bucle que tenga cinco o más instrucciones ECNB, la primera instrucción ECNB podría tener un índice de "000" y la cuarta instrucción de no bifurcación idónea podría tener un índice de "011". El índice representa una dirección dentro de la CHT 312 para acceder a los valores almacenados de contador de estado de ejecución para la instrucción ECNB correspondiente.

15 **[0033]** El circuito de predicción 314 recibe la información de predicción para una instrucción ECNB, tales como valores de salida de contador de estado de ejecución, y predice, durante la etapa de descodificación 231 de la FIG. 2, por ejemplo, que la instrucción ECNB no se ejecutará. En un modo de realización alternativo, el circuito de predicción 314 puede predecir que la condición especificada por la instrucción ECNB se evalúa en un estado de no ejecución. El circuito de predicción 314 pasa la decisión de predicción al circuito de rastreo 316, que puede incluir la instrucción ECNB asociada que se está prediciendo y contenidos de entrada de CHT correspondientes. Si no se predice una instrucción ECNB, la información de predicción indica una ejecución regular. Si se predijo que una instrucción ECNB se va a ejecutar como una instrucción NOP, entonces la información de rastreo informa al circuito de corrección 318 sobre el estado de ejecución y la evaluación de la condición asociada para determinar si se realizó una predicción incorrecta. Si se realizó una predicción incorrecta, el circuito de corrección 318 inicializa el procesamiento en cadena, actualiza los contadores adecuados de estado de ejecución de la CHT 312 y, en un modo de realización, marca la entrada de CHT asociada para indicar que esta instrucción ECNB particular no se ha de predecir a partir de este momento. En otro modo de realización, el circuito de corrección 318 también puede cambiar el criterio de evaluación preestablecido al determinar que la instrucción ECNB fue mal predicha, por ejemplo, para hacer que el criterio de predicción sea más conservador a partir de este momento.

35 **[0034]** Se reconoce que una secuencia de instrucciones de no bifurcación condicional idónea (ECNB) en un bucle puede codificarse de tal manera que cada instrucción dependa de la misma resolución de condición. En tal caso, la secuencia de instrucciones ECNB se puede tratar como un grupo con una sola entrada en una tabla de historial condicional (CHT). En tal caso, cuando la predicción indica que no hay ejecución, la secuencia de instrucciones ECNB se trata como una secuencia de instrucciones de no operación (NOP). Por ejemplo, un grupo de instrucciones ECNB puede incluir dos instrucciones de operandos de carga condicional seguidas de una instrucción aritmética condicional que especifica una operación en los dos operandos cargados. Además, las tres instrucciones ECNB dependen de la misma resolución de condición. En un procesador de procesamiento en cadena, estas tres instrucciones pueden identificarse antes en el procesamiento en cadena como un grupo condicional que tiene la misma resolución de condición. En un modo de realización, la primera instrucción de carga condicional del grupo en procesamiento en cadena activa una evaluación de predicción, y una entrada en la CHT puede marcarse como asociada a este grupo de instrucciones ECNB. De esta manera, el grupo de instrucciones de ECNB se asocia a un índice único en la CHT para que todas las instrucciones de un grupo ECNB se evalúen con respecto al mismo índice.

50 **[0035]** Se reconoce que las instrucciones de no bifurcación condicional idónea (ECNB) pueden reconocerse fuera de los bucles y también se puede predecir de forma ventajosa que no se ejecuten. El circuito de detección 304, que actúa como un circuito de detección de intervalo de direcciones, detecta un intervalo de direcciones donde se evaluará la predicción de instrucciones ECNB. Cuando se extrae código que está dentro del intervalo de direcciones, se habilita el circuito de predicción de instrucciones ECNB 300 y se supervisan y evalúan las instrucciones ECNB dentro del intervalo de direcciones. Cuando se cumple un criterio de evaluación, se predice que la instrucción ECNB se ejecutará o no se ejecutará con un funcionamiento de rastreo y corrección de manera similar a lo descrito anteriormente.

55 **[0036]** Además, se reconoce que no todos los bucles o intervalos de direcciones tienen características similares. Si un bucle o un intervalo de direcciones particular proporciona malos resultados de predicción, ese bucle o intervalo de direcciones puede marcarse para inhabilitar la predicción. De forma similar, un bucle o un intervalo de direcciones particular puede funcionar con buena predicción en un conjunto de escenarios operativos y puede funcionar con mala predicción en un conjunto diferente de escenarios operativos. En tal caso, el reconocimiento de los escenarios operativos permite que la predicción se habilite, se inhabilite o se habilite pero con un criterio de evaluación diferente adecuado para el escenario operativo.

65 **[0037]** La FIG. 4A ilustra un primer proceso 400 para predecir la ejecución de una instrucción ECNB. En el bloque 402, la ejecución de código de procesador se supervisa para un bucle de software. En el bloque de decisión 404, se determina si se ha alcanzado un punto en el código donde se detectó un bucle de software. Un bucle de software

puede determinarse, por ejemplo, identificando una bifurcación hacia atrás hacia un inicio de un bucle, como se describió anteriormente. Si no se ha identificado ningún bucle de software, el primer proceso 400 vuelve al bloque 402. Si se ha identificado un bucle de software, entonces, en este punto del código, ya se ha ejecutado un primer ciclo del bucle de software y el siguiente ciclo del bucle de software está listo para comenzar.

5
[0038] En el siguiente ciclo del bucle de software, en el bloque de decisión 406, se determina si se ha detectado una instrucción ECNB, por ejemplo, durante una etapa de descodificación del procesamiento en cadena, tal como la etapa de descodificación 231 de la FIG. 2. Si no se ha detectado ninguna instrucción ECNB, el proceso 400 continúa con el bloque de decisión 408. En el bloque de decisión 408 se determina si se ha completado una pasada a través del bucle de software. Una primera pasada a través del bucle de software puede determinarse, por ejemplo, llegando a la bifurcación hacia atrás que identificó el bucle de software en el bloque de decisión 404. Si no se ha completado una pasada a través del bucle de software, el primer proceso 400 vuelve al bloque de decisión 406 para continuar la verificación de una instrucción ECNB. En el bloque de decisión 406, si se ha detectado una instrucción ECNB, el primer proceso 400 continúa con el bloque de decisión 410. En el bloque de decisión 410 se determina, durante la etapa de descodificación de procesador 231, por ejemplo, si se ha cumplido con un criterio de evaluación preestablecido para esta instrucción ECNB. El criterio de evaluación preestablecido puede ser, por ejemplo, si un conteo de iteraciones de bucle es mayor o igual que un valor preestablecido, tal como tres. Si no se ha cumplido el criterio de evaluación preestablecido, el primer proceso 400 continúa con el bloque 412. En el bloque 412, esta instrucción ECNB se ejecuta y un estado de ejecución se actualiza para esta instrucción ECNB. Por ejemplo, un indicador de predicción de inhabilitación se establece si la instrucción ECNB se ejecuta de forma condicional. Un indicador de predicción de inhabilitación una vez establecido puede no reajustarse, por ejemplo, hasta que se complete el bucle de software.

25
[0039] En el bloque de decisión 408 se determina si se ha completado una pasada a través del bucle de software. Si se ha completado una pasada a través del bucle de software, el primer proceso 400 continúa con el bloque de decisión 414. En el bloque de decisión 414 se determina si el bucle de software ha finalizado. Si el bucle de software no ha finalizado, el primer proceso 400 continúa con el bloque 416. En el bloque 416 se cuenta la iteración de bucle, y el primer proceso 400 vuelve al bloque de decisión 406 para seguir verificando las instrucciones ECNB. Si el bucle de software finaliza, el primer proceso 400 continúa con el bloque 418. En el bloque 418, los circuitos de predicción usados en el primer proceso 400 se reajustan. Tal reajuste permite que la evaluación de la predicción comience con circuitos reinicializados cada vez que se entre en un bucle de software. De forma alternativa, el reajuste podría ocurrir siempre que se detecte un nuevo bucle de software. El primer proceso 400 vuelve después al bloque 402 para empezar a buscar el siguiente bucle de software.

35
[0040] Volviendo al bloque de decisión 410, si el criterio preestablecido se ha cumplido, el primer proceso 400 continúa con el bloque de decisión 420. En el bloque de decisión 420 se determina si se cumple una condición de ejecución para esta instrucción ECNB. Por ejemplo, una condición de ejecución puede tomar la forma de un indicador de inhabilitación de predicción para esta instrucción ECNB. Un indicador de inhabilitación de predicción generalmente se establecería cada vez que se ejecute de forma condicional una instancia de la instrucción ECNB. Un indicador de inhabilitación de predicción una vez establecido puede no reajustarse, por ejemplo, hasta que se complete el bucle de software. Volviendo al bloque de decisión 420, si el indicador de inhabilitación de predicción está en el estado de predicción inhabilitada que indica que la instrucción ECNB se ha ejecutado alguna vez, el primer proceso 400 vuelve al bloque 412. Si el indicador de inhabilitación de predicción está en el estado de predicción habilitada que indica que la instrucción ECNB no se ha ejecutado previamente, el primer proceso 400 continúa con el bloque 421. En el bloque 421, se predice que esta instrucción ECNB se ejecutará como una instrucción NOP. En el bloque 422, la predicción se rastrea en el procesamiento en cadena del procesador. En el bloque de decisión 424 se determina, en la etapa de procesamiento en cadena donde se determina la condición asociada con esta instrucción ECNB, si la predicción del bloque 420 fue correcta. Si la predicción fue correcta, el proceso 400 vuelve al bloque 408, ya que puede ser necesario evaluar otras instrucciones ECNB en el bucle de software. Si la predicción fue incorrecta, el primer proceso 400 continúa con el bloque 426. En el bloque 426 se lleva a cabo un reinicio del procesamiento en cadena de procesador para eliminar la instrucción ECNB predicha incorrectamente y cualquier instrucción en el procesamiento en cadena que pueda haber sido afectada por la operación predicha. En el bloque 426, el procesamiento en cadena se corrige hasta el punto de detectar esta instrucción ECNB. El proceso 400 regresa después al bloque 412, donde esta instrucción ECNB se puede ejecutar y su estado de ejecución asociado se puede actualizar.

60
[0041] La FIG. 4B ilustra un segundo proceso 450 para predecir la ejecución de una instrucción ECNB. En el bloque 452, la ejecución de código de procesador se supervisa para detectar una instrucción ECNB. En bloque de decisión 454 se determina si se ha detectado una instrucción ECNB, por ejemplo, durante una etapa de descodificación del procesamiento en cadena, tal como una etapa de descodificación 231 de la FIG. 2. Si no se ha detectado ninguna instrucción ECNB, el segundo proceso 450 vuelve al bloque 452. Si se ha detectado una instrucción ECNB, el segundo proceso 450 continúa con el bloque de decisión 456. En el bloque de decisión 456 se determina, durante la etapa de descodificación de procesador 231, por ejemplo, si se ha cumplido con un criterio de evaluación preestablecido para esta instrucción ECNB. El criterio de evaluación preestablecido puede ser, por ejemplo, si un conteo de iteración de bucle asociado a la instrucción ECNB es mayor que o igual a un valor preestablecido, tal como tres. Si el criterio de evaluación preestablecido no se ha cumplido, el segundo proceso

450 continúa con el bloque 458. En el bloque 458, esta instrucción ECNB se ejecuta y un contador de estado de ejecución se actualiza para esta instrucción ECNB.

5 **[0042]** En el bloque de decisión 460 se determina si se ha detectado un bucle de software. Un bucle de software se puede determinar, por ejemplo, identificando una bifurcación hacia atrás en el código, como se describe anteriormente. Si no se ha detectado un bucle de software, el segundo proceso 450 vuelve al bloque 452 para verificar otra instrucción ECNB. Si se ha identificado un bucle de software, el segundo proceso 450 continúa con el bloque 462. En el bloque 462, los contadores de estado de ejecución para las instrucciones ECNB que no forman parte del bucle detectado se inicializan, ya que en el segundo proceso 450, solo se predicen instrucciones ECNB en un bucle de software. La FIG. 4B cubre un caso esperado en el que un bucle se detecta en una secuencia de código que tiene otras instrucciones ECNB fuera del bucle. Las otras instrucciones ECNB fuera del bucle afectan a la capacidad de CHT y podrían limitar el número de instrucciones ECNB evaluadas en el bucle detectado. Por lo tanto, los contadores de estado de ejecución de las instrucciones ECNB encontradas fuera del bucle se reinician y la lógica CHT se ajusta como se describe con más detalle a continuación.

15 **[0043]** Las instrucciones ECNB que no forman parte del bucle de software detectado se pueden determinar a partir de las direcciones de las instrucciones ECNB y el intervalo de direcciones del bucle de software. La entrada inicial de una tabla de historial condicional (CHT) se ajusta para representar las instrucciones ECNB detectadas en el bucle de software. También cabe señalar que los contadores de estado de ejecución para las instrucciones ECNB que no forman parte del bucle detectado se pueden reasignar a la CHT para aumentar la capacidad de la CHT para instrucciones ECNB dentro del bucle de software. En el bloque de decisión 464 se determina si el bucle de software ha finalizado. Si el bucle de software no ha finalizado, el segundo proceso 450 continúa con el bloque 466. En el bloque 466 se cuenta la iteración de bucle y el proceso vuelve al bloque 452. Si el bucle de software ha finalizado, el segundo proceso 450 continúa con el bloque 468. En el bloque 468, los circuitos de predicción usados en el primer proceso 400 se reajustan. Dicho reajuste permite que, cada vez que se entre en un bucle de software, la evaluación de predicción comience con los circuitos reinicializados. De forma alternativa, el reajuste podría ocurrir siempre que se detecte un nuevo bucle de software.

20 **[0044]** Volviendo al bloque de decisión 456, si el criterio preestablecido se ha cumplido, el segundo proceso 450 continúa con el bloque de decisión 470. En el bloque de decisión 470 se determina si ejecutar esta instrucción ECNB como una instrucción de no operación (NOP). Por ejemplo, se puede predecir que esta instrucción ECNB ejecuta la función especificada por la instrucción ECNB. En tal caso, el segundo proceso 450 continúa con el bloque 458. De forma alternativa, se puede predecir que esta instrucción ECNB se ejecute como una instrucción NOP. En el bloque 472, la predicción se rastrea en el procesamiento en cadena del procesador. En el bloque de decisión 474 se determina, en la etapa de procesamiento en cadena donde se determina la condición asociada con esta instrucción ECNB, si la predicción del bloque 470 fue correcta. Si la predicción fue correcta, el segundo proceso 450 vuelve al bloque 460. Si la predicción fue incorrecta, el segundo proceso 450 continúa con el bloque 476. En el bloque 476 se lleva a cabo un reinicio del procesamiento en cadena de procesador para eliminar la instrucción ECNB predicha incorrectamente y cualquier instrucción en el procesamiento en cadena que pueda haber sido afectada por la operación predicha. En el bloque 478, los circuitos de predicción usados en el segundo proceso 450 se reajustan, debido a que se encuentra una predicción incorrecta en el bucle de software que se está evaluando. El segundo proceso 450 vuelve entonces al bloque 452. De forma alternativa, se podría realizar una corrección en los contadores de estado de instrucción ECNB para reflejar la predicción incorrecta y el proceso puede continuar.

30 **[0045]** La FIG. 5 ilustra un tercer proceso 500 para predecir la ejecución de una instrucción ECNB. En el bloque 502, la ejecución de código de procesador se supervisa para determinar si el procesador está ejecutando código extraído de un intervalo de direcciones previamente especificado. Por ejemplo, un compilador u otra herramienta de software puede identificar instrucciones ECNB en una sección de código y usar las direcciones de las instrucciones ECNB identificadas para generalizar un intervalo de direcciones preestablecido. En bloque de decisión 504 se determina si se ha detectado el intervalo de direcciones preestablecido, por ejemplo, durante una etapa de extracción del procesamiento en cadena, tal como una etapa de extracción 230 de la FIG. 2. Si no se ha detectado ningún intervalo de direcciones preestablecido, el tercer proceso 500 vuelve al bloque 502. Si se ha detectado el intervalo de direcciones preestablecido, el tercer proceso 500 continúa con el bloque 506. En el bloque 506 se actualiza un contador de intervalo de direcciones para indicar el número de veces que se ha entrado en un intervalo de direcciones particular. En el bloque 508, el código de procesador se supervisa para detectar una instrucción ECNB. En bloque de decisión 510 se determina si se ha detectado una instrucción ECNB, por ejemplo, durante una etapa de decodificación del procesamiento en cadena, tal como una etapa de decodificación 231 de la FIG. 2. Si no se ha detectado ninguna instrucción ECNB, el tercer proceso 500 continúa con el bloque de decisión 512. En el bloque de decisión 512 se determina si el procesador todavía está ejecutando código en el intervalo de direcciones preestablecido. Si el procesador no está ejecutando código en el intervalo de direcciones preestablecido, el tercer proceso 500 continúa con el bloque 502. Si el procesador está ejecutando código en el intervalo de direcciones preestablecido, el tercer proceso 500 continúa con el bloque 508.

35 **[0046]** Volviendo al bloque de decisión 510, si se ha detectado una instrucción ECNB, el tercer proceso 500 continúa con el bloque de decisión 514. En el bloque de decisión 514 se determina, durante la etapa de

descodificación de procesador 231 de la FIG. 2, por ejemplo, si se ha cumplido con un criterio de evaluación preestablecido para esta instrucción ECNB. Se elige un criterio de evaluación preestablecido para proporcionar un alto nivel de confianza para predecir que la instrucción ECNB se ejecuta como una NOP. Por ejemplo, en un modo de realización, el criterio de evaluación preestablecido se puede configurar para requerir que al menos dos intentos previos de ejecución de la instrucción ECNB tengan un estado fuertemente no ejecutado. Si el criterio de evaluación preestablecido no se ha cumplido, el tercer proceso 500 continúa con el bloque 516. En el bloque 516, esta instrucción ECNB se ejecuta y un contador de estado de ejecución se actualiza para esta instrucción ECNB. El tercer proceso 500 vuelve después al bloque de decisión 512 para determinar si el procesador sigue ejecutando código en el intervalo de direcciones preestablecido y vuelve al bloque 508 si la determinación es positiva y, de lo contrario, vuelve al bloque 502.

[0047] Volviendo al bloque de decisión 514, si se ha cumplido con el criterio de evaluación preestablecido, el tercer proceso 500 continúa con el bloque 520. En el bloque 520, se predice que la ejecución de esta instrucción ECNB se ejecutará como una instrucción NOP. En el bloque 522, la predicción se rastrea en el procesamiento en cadena del procesador. En el bloque de decisión 524 se determina, en la etapa de procesamiento en cadena donde se determina la condición asociada con esta instrucción ECNB, si la predicción del bloque 520 fue correcta. Si la predicción fue correcta, el tercer proceso 500 vuelve al bloque de decisión 512 para determinar si el procesador sigue ejecutando código en el intervalo de direcciones preestablecido y vuelve al bloque 508 si la determinación es positiva y, de lo contrario, vuelve al bloque 502.

[0048] Volviendo al bloque de decisión 524, si la predicción fue incorrecta, el tercer proceso 500 continúa con el bloque 528. En el bloque 528 se lleva a cabo un reinicio del procesamiento en cadena de procesador para eliminar la instrucción ECNB predicha incorrectamente y cualquier instrucción en el procesamiento en cadena que pueda haber sido afectada por la operación predicha. En el bloque 530 se actualizan los circuitos de predicción para esta instrucción ECNB. El proceso 500 vuelve entonces al bloque 508.

[0049] La FIG. 6 ilustra un cuarto proceso 600 para predecir la ejecución de una instrucción ECNB. El cuarto proceso 600 evalúa si una instrucción ECNB se identifica repetidamente por tener un período relativamente corto o relativamente largo de ciclos de procesador entre la identificación de la misma instrucción ECNB. Un período relativamente corto de ciclos de procesador puede ser indicativo de que la instrucción ECNB se encuentra en un bucle de software. Un período relativamente largo de ciclos de procesador puede ser indicativo de que la instrucción ECNB está ubicada dentro de un intervalo de direcciones que puede ejecutarse principalmente debido a una rutina llamada, como cuando un usuario descarga un vídeo para su visualización. En tal caso, se puede llamar a una rutina de descodificación MPEG, que tiene una instrucción ECNB.

[0050] En el bloque 602, la ejecución de código de procesador se supervisa para detectar una instrucción ECNB. En el bloque de decisión 604 se determina si se ha detectado una instrucción ECNB, por ejemplo, durante una etapa de descodificación del procesamiento en cadena, tal como una etapa de descodificación 231 de la FIG. 2. Si no se ha detectado ninguna instrucción ECNB, el cuarto proceso 600 vuelve al bloque 602. Si se ha detectado una instrucción ECNB, el cuarto proceso 600 continúa con el bloque de decisión 606. En el bloque de decisión 606 se determina si se ha identificado previamente esta instrucción ECNB. Si esta es la primera vez que se ha identificado esta instrucción ECNB, el cuarto proceso 600 continúa con el bloque 608. En el bloque 608 se registra la dirección de esta instrucción ECNB. En el bloque 610, se inicia un contador de correspondencias para, por ejemplo, un conteo de uno. En el bloque 612, un contador de ciclos transcurridos comienza a contar el número de ciclos transcurridos entre apariciones de esta instrucción ECNB. Se observa que puede ser necesario filtrar el número de ciclos contados para tener en cuenta rutinas de interrupción y operaciones de acceso directo a memoria en la medida en que los ciclos asociados a estas otras operaciones afecten a la precisión del conteo para el propósito previsto. En el bloque 614, esta instrucción ECNB se ejecuta y un contador de estado de ejecución se actualiza. El cuarto proceso 600 vuelve entonces al bloque 602.

[0051] Volviendo al bloque de decisión 606, si esta instrucción ECNB ha sido identificada previamente, entonces el cuarto proceso 600 continúa con el bloque 618. En el bloque 618 se evalúa la cantidad de veces que se ha encontrado esta instrucción ECNB y la cantidad de ciclos transcurridos entre apariciones. En el bloque 619, el contador de correspondencias se actualiza, el conteo de ciclos transcurridos actual se almacena, y el contador de ciclos transcurridos se reinicia para contar el número de ciclos que transcurren en el siguiente período entre apariciones. En el bloque de decisión 620, se determina si se evalúa un criterio de evaluación preestablecido. En un modo de realización, el criterio de evaluación preestablecido se puede configurar para requerir que al menos dos intentos de ejecución previos tengan un estado fuertemente no ejecutado en un contador de estado de ejecución con menos de X ciclos de procesador entre las dos apariciones. En otro modo de realización, el criterio de evaluación preestablecido se puede configurar para requerir al menos tres intentos de ejecución previos, cada uno teniendo un estado fuertemente no ejecutado en el contador de estado de ejecución, con al menos Y ciclos de procesador entre cada una de las tres apariciones, donde Y es mayor que X. Si no se cumple el criterio de evaluación preestablecido, el cuarto proceso 600 vuelve al bloque 614 donde se ejecuta esta instrucción ECNB y se actualiza el contador de estado de ejecución. El proceso vuelve después al bloque 602.

[0052] Volviendo al bloque de decisión 620, si se cumple el criterio de evaluación preestablecido, el cuarto proceso 600 continúa con el bloque 624. En el bloque 624, se predice la ejecución de esta instrucción ECNB; por ejemplo, se predice que esta instrucción ECNB se ejecutará como una instrucción NOP. En el bloque 626, la predicción se rastrea en el procesamiento en cadena del procesador. En el bloque de decisión 628 se determina, en la etapa de procesamiento en cadena donde se determina la condición asociada con esta instrucción ECNB, si la predicción del bloque 624 fue correcta. Si la predicción fue correcta, el cuarto proceso 600 vuelve al bloque 602. Si la predicción fue incorrecta, el cuarto proceso 600 continúa con el bloque 632. En el bloque 632 se lleva a cabo un reinicio del procesamiento en cadena de procesador para eliminar la instrucción ECNB predicha incorrectamente y cualquier instrucción en el procesamiento en cadena que pueda haber sido afectada por la operación predicha. En el bloque 634, el circuito de predicción usado para esta instrucción ECNB se reajusta. El proceso 600 vuelve entonces al bloque 602.

[0053] Los diversos bloques lógicos, módulos, circuitos, elementos y/o componentes ilustrativos descritos en relación con los modos de realización dados a conocer en el presente documento pueden implementarse o realizarse con un procesador de propósito general, con un procesador de señales digitales (DSP), con un circuito integrado específico de la aplicación (ASIC), con una formación de puertas programables in situ (FPGA) o con otros componentes de lógica programable, lógica de transistor o de puertas discretas, componentes de hardware discretos, o con cualquier combinación de los mismos diseñada para realizar las funciones descritas en el presente documento. Un procesador de propósito general puede ser un microprocesador pero, como alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también puede implementarse como una combinación de componentes informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo, adecuada para una aplicación deseada.

[0054] Los procedimientos descritos en relación con los modos de realización divulgadas en el presente documento pueden realizarse directamente en hardware, en un módulo de software ejecutado por un procesador o en una combinación de los dos. Un módulo de software puede residir en una memoria RAM, en una memoria flash, en una memoria ROM, en una memoria EPROM, en una memoria EEPROM, en registros, en un disco duro, en un disco extraíble, en un CD-ROM o en cualquier otra forma de medio de almacenamiento conocido en la técnica. Un medio de almacenamiento puede estar acoplado al procesador de manera que el procesador pueda leer información de, y escribir información en, el medio de almacenamiento. Como alternativa, el medio de almacenamiento puede estar integrado en el procesador.

[0055] El procesador 210, por ejemplo, puede configurarse para ejecutar instrucciones que incluyen instrucciones de no bifurcación condicional, bajo el control de un programa almacenado en un medio de almacenamiento legible por ordenador ya sea directamente asociado localmente al procesador, tal como puede estar disponible mediante una memoria caché de instrucciones, o bien accesible a través de un dispositivo de E/S, tal como uno de los dispositivos de E/S 240 o 242, por ejemplo. El dispositivo de E/S también puede acceder a datos que residen en un dispositivo de memoria, ya sea directamente asociado localmente a los procesadores, tales como la Dcaché 228, o bien accesible desde la memoria de otro procesador. El medio de almacenamiento legible por ordenador puede incluir memoria de acceso aleatorio (RAM), memoria de acceso aleatorio dinámica (DRAM), memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria flash, memoria de solo lectura (ROM), memoria programable de solo lectura (PROM), memoria de solo lectura programable borrable (EPROM), memoria de solo lectura programable y borrable eléctricamente (EEPROM), disco compacto (CD), disco de vídeo digital (DVD), otros tipos de discos extraíbles o cualquier otro medio de almacenamiento adecuado.

[0056] Si bien la invención se divulga en el contexto de modos de realización ilustrativos para su uso en sistemas de procesador, se reconocerá que una amplia variedad de implementaciones puede ser empleada por expertos en la técnica en consonancia con el análisis anterior y las reivindicaciones que siguen a continuación. Por ejemplo, una implementación de función fija también puede utilizar varios modos de realización de la presente invención.

REIVINDICACIONES

1. Un procedimiento para manejar una instrucción de no bifurcación condicional, que comprende:
 - 5 identificar (406, 454, 510, 604) una instrucción de no bifurcación condicional, CNB, como idónea para una predicción;
 - predecir, en respuesta a satisfacer un criterio de evaluación (410, 456, 510), si la instrucción CNB idónea se ejecutará o no, en el que el criterio de evaluación de predicción corresponde a un estado fuertemente no ejecutado;
 - 10 ejecutar la instrucción CNB idónea como una instrucción de no operación, NOP (420, 520) en respuesta a una predicción de que la instrucción CNB idónea no se ejecutará; de forma alternativa, ejecutar con normalidad la instrucción CNB idónea en respuesta a una predicción de que la instrucción CNB idónea va a ejecutarse; y
 - 15 registrar en una tabla de historial condicional (252, 312) si la instrucción CNB idónea se ejecutó o no y predecir que la siguiente instrucción CNB idónea no se ejecuta en respuesta a la tabla de historial condicional que indica que al menos un intento de ejecución anterior de la instrucción CNB idónea no se ejecutó,
 - 20 **caracterizado por que** el al menos un intento de ejecución anterior de la instrucción CNB idónea se encontró en un bucle de software, en un intervalo de direcciones preestablecido o dentro de un número identificado de ciclos de procesador.
 - 25
2. El procedimiento de la reivindicación 1, en el que:
 - un operando de origen requerido para que la instrucción CNB idónea se ejecute no se extrae en respuesta a la predicción de que la instrucción CNB idónea no se ejecutará; o
 - 30 un registro en un archivo de registro de propósito general no está reservado para contener el resultado de la instrucción CNB idónea en respuesta a la predicción de que la instrucción CNB idónea no se ejecutará.
- 35 3. El procedimiento de la reivindicación 1, que comprende además:
 - predecir que la instrucción CNB idónea no se ejecuta en respuesta a un indicador de inhabilitación de predicción que indica que no se realizaron ejecuciones exitosas anteriores de la instrucción CNB idónea durante un período idóneo en el que se basó la predicción.
 - 40
4. El procedimiento de la reivindicación 1, que comprende además:
 - comparar (410, 456, 514, 620) el criterio de evaluación con una salida de valor de conteo de un contador de estado de ejecución de instrucción CNB idónea para generar la predicción, en el que el contador de estado de ejecución de instrucción CNB idónea se satura en un primer valor de conteo indicativo de un historial de intentos de ejecución anteriores de la instrucción CNB idónea como fuertemente no ejecutada.
 - 45
5. El procedimiento de la reivindicación 4, que comprende además:
 - 50 actualizar el contador de estado de ejecución de instrucción CNB idónea en un primer sentido para indicar un intento de ejecución anterior de la instrucción CNB idónea ejecutada de forma condicional; y
 - actualizar el contador de estado de ejecución de instrucción CNB idónea en un segundo sentido que es opuesto al primer sentido para indicar que un intento de ejecución anterior de la instrucción CNB idónea no se ejecutó de forma condicional.
 - 55
6. El procedimiento de la reivindicación 4, en el que el criterio de evaluación es el primer valor de conteo.
7. Un aparato de procesamiento en cadena de instrucciones (220), que comprende:
 - 60 medios (250, 310) para identificar una instrucción de no bifurcación condicional como idónea para una predicción;
 - medios (254, 314) para predecir si se ejecutará o no la instrucción de no bifurcación condicional idónea, CNB idónea, en el que el criterio de evaluación de predicción corresponde a un estado fuertemente no ejecutado como se registra en una tabla de historial condicional (252, 312); y
 - 65

medios (233) para ejecutar la instrucción CNB idónea como una instrucción de no operación, NOP, en respuesta a una predicción de que la instrucción CNB idónea no se ejecutará, o para ejecutar con normalidad la instrucción CNB idónea en respuesta a una predicción de que la instrucción CNB idónea va a ejecutarse;

caracterizado por un medio para predecir que la siguiente instrucción CNB idónea no se ejecuta en respuesta a la tabla de historial condicional que indica que al menos un intento de ejecución anterior de la instrucción CNB idónea no se ejecutó, cuando el al menos un intento de ejecución anterior de la instrucción CNB idónea se encontró en un bucle de software, en un intervalo de direcciones preestablecido o dentro de un número identificado de ciclos de procesador.

8. El aparato de la reivindicación 7, en el que:

dichos medios de identificación comprenden un primer circuito para identificar la instrucción de no bifurcación condicional como idónea para una predicción; y

dichos medios de predicción comprenden un segundo circuito para predecir si la instrucción CNB idónea se ejecutará o no en respuesta al cumplimiento del criterio de evaluación.

9. El aparato de la reivindicación 8, que comprende además:

un circuito de extracción de operando configurado para no extraer un operando requerido para que la instrucción CNB idónea se ejecute en respuesta a la predicción de no ejecución.

10. El aparato de la reivindicación 8, que comprende además:

un circuito de rastreo de procesamiento en cadena configurado para rastrear información de predicción en etapas de procesamiento en cadena después de una etapa de predicción del procesamiento en cadena; y

un circuito de etapa de ejecución de instrucción CNB idónea configurado para no ejecutar la instrucción CNB idónea en respuesta a la predicción de no ejecución.

11. El aparato de la reivindicación 8, que comprende además:

un contador de estado de ejecución de instrucción CNB idónea con una salida de valor de conteo que se compara con el criterio de evaluación;

medios para actualizar el valor de conteo en una primera dirección para indicar una instrucción CNB idónea ejecutada de forma condicional, en el que el valor de conteo se satura en un primer valor de conteo indicativo de un historial fuertemente ejecutado; y

medios para actualizar el valor de conteo en una segunda dirección para indicar que una instrucción CNB idónea no se ejecutó, en el que el valor de conteo se satura en un segundo valor de conteo indicativo de un historial fuertemente no ejecutado.

12. El aparato de la reivindicación 11, en el que el criterio de evaluación es el segundo valor de conteo.

13. El aparato de la reivindicación 8, en el que el criterio de evaluación es un indicador de inhabilitación de predicción en un estado no activo, y en el que el estado no activo del indicador de inhabilitación de predicción indica que la predicción está habilitada, comprendiendo el aparato medios para establecer el indicador de inhabilitación de predicción en un estado de inhabilitación si alguna vez se determina que la instrucción CNB idónea se ha ejecutado de forma condicional en un bucle de software asociado a la instrucción CNB idónea.

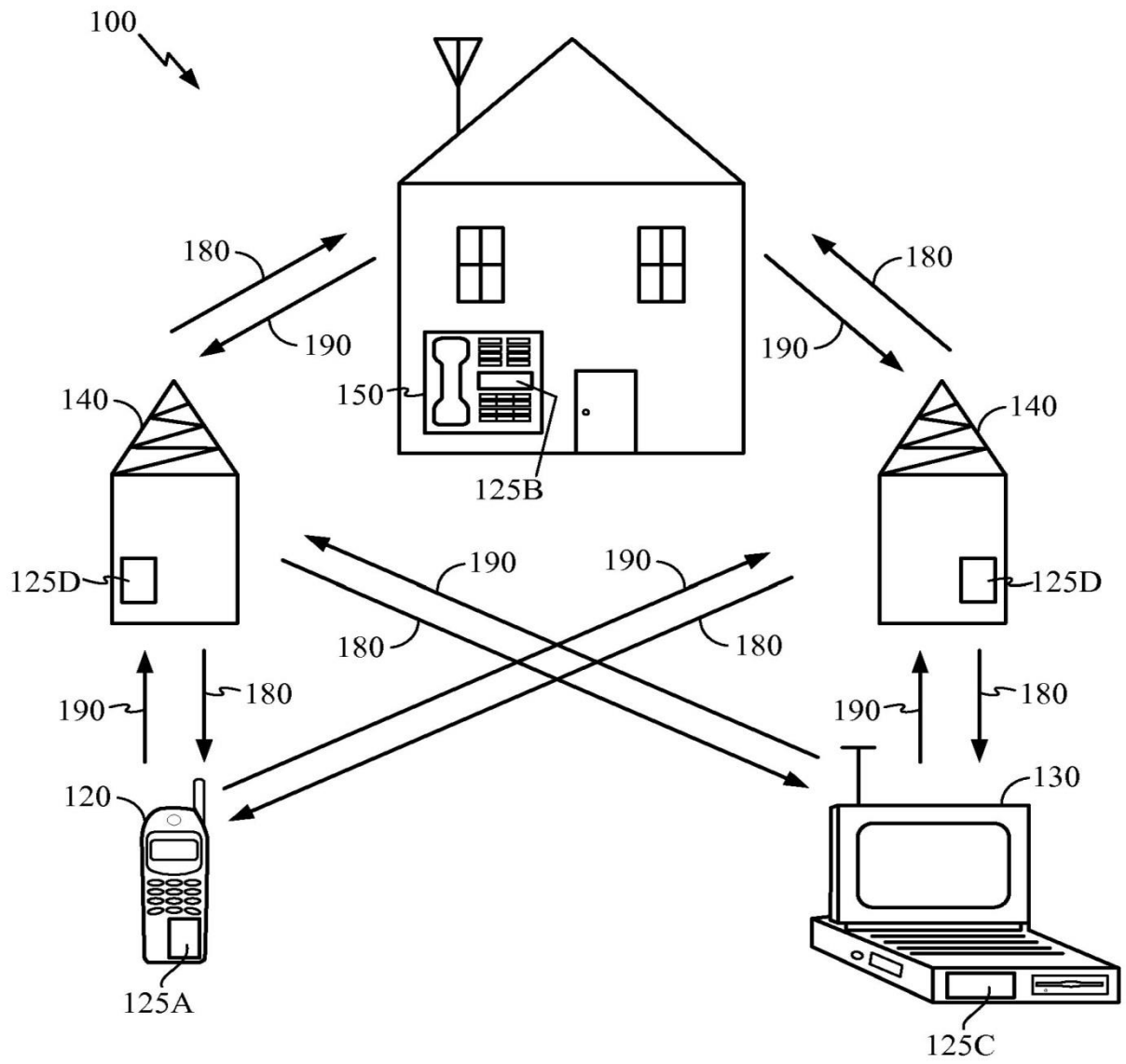


FIG. 1

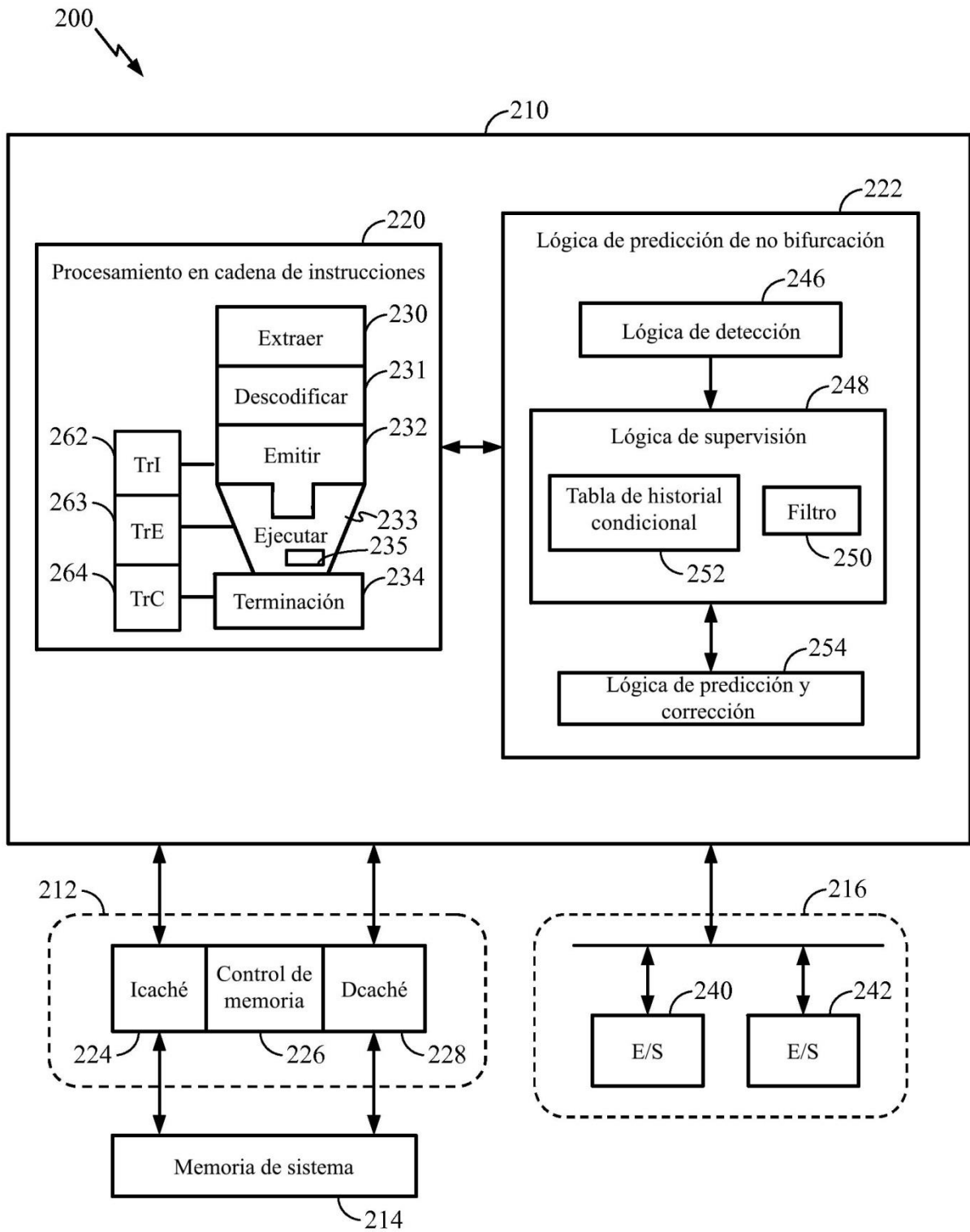


FIG. 2

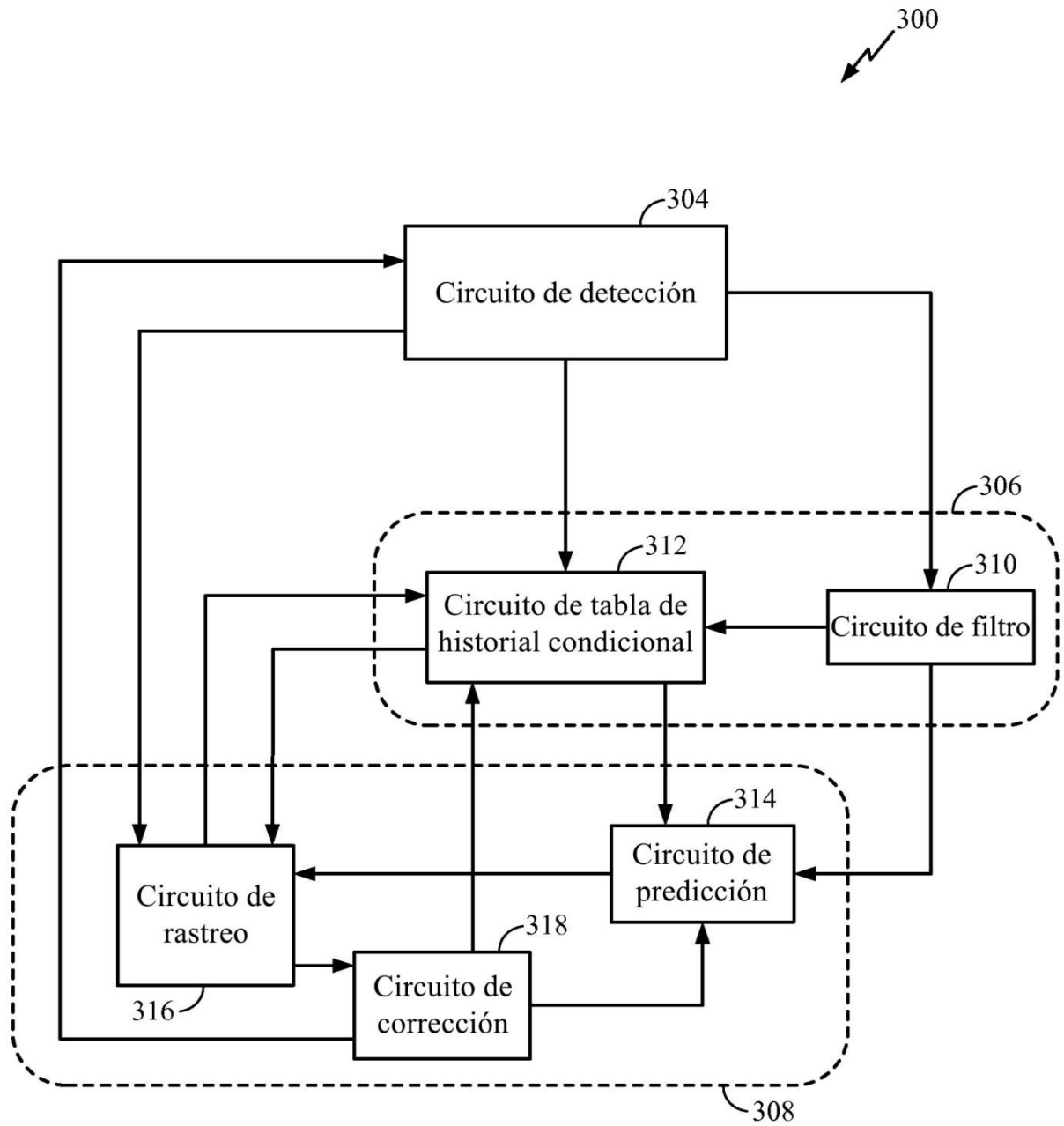


FIG. 3

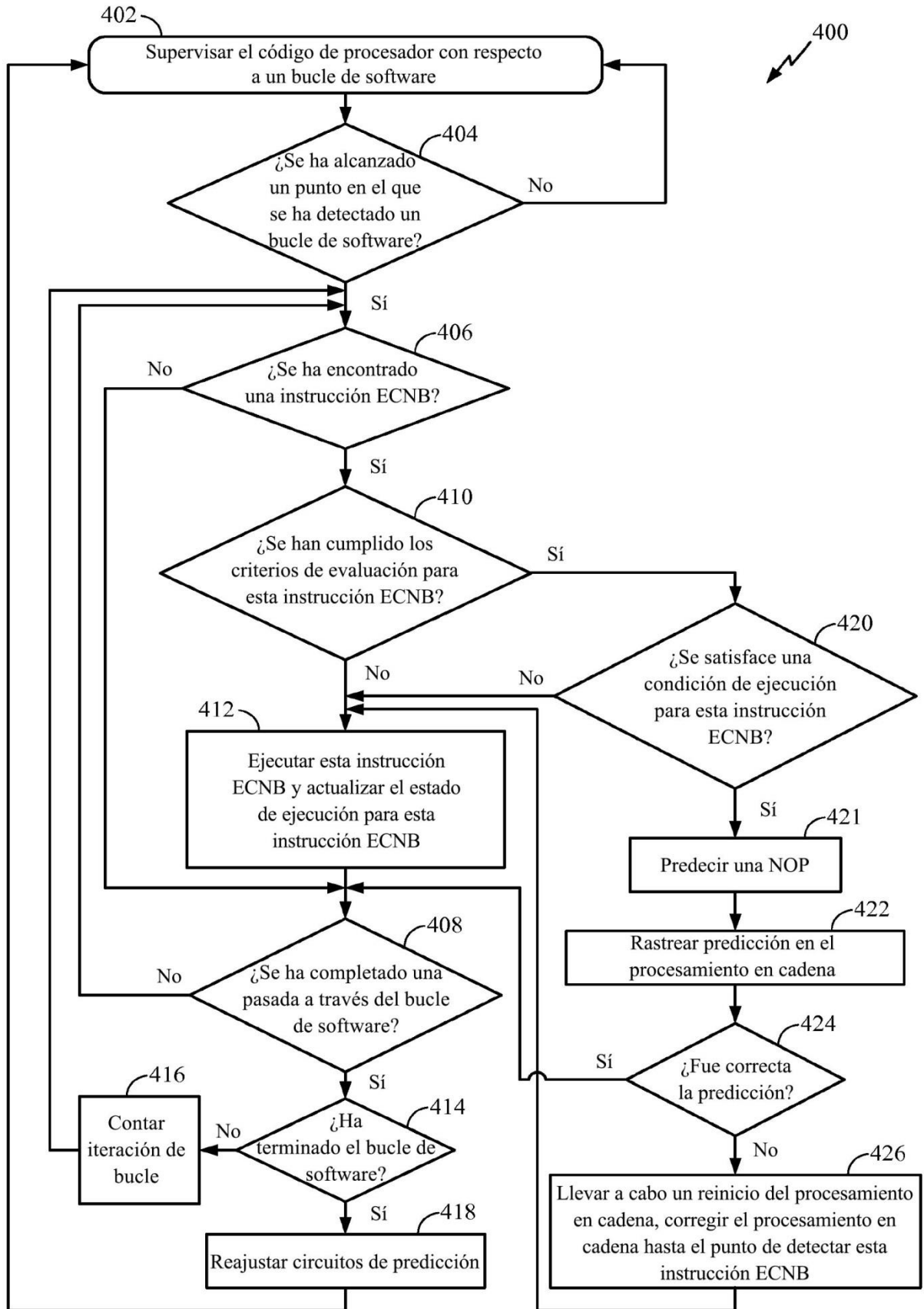


FIG. 4A

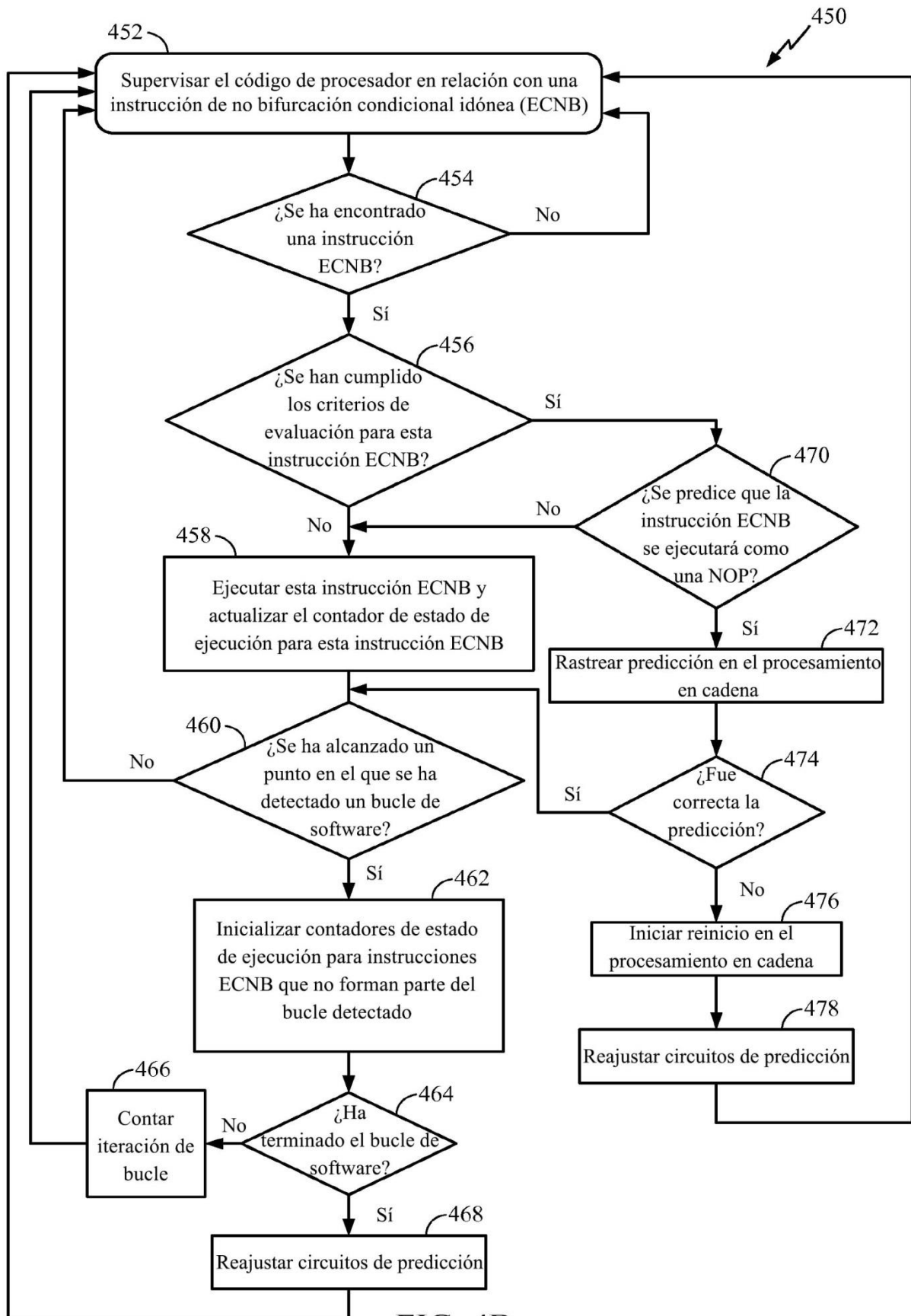


FIG. 4B

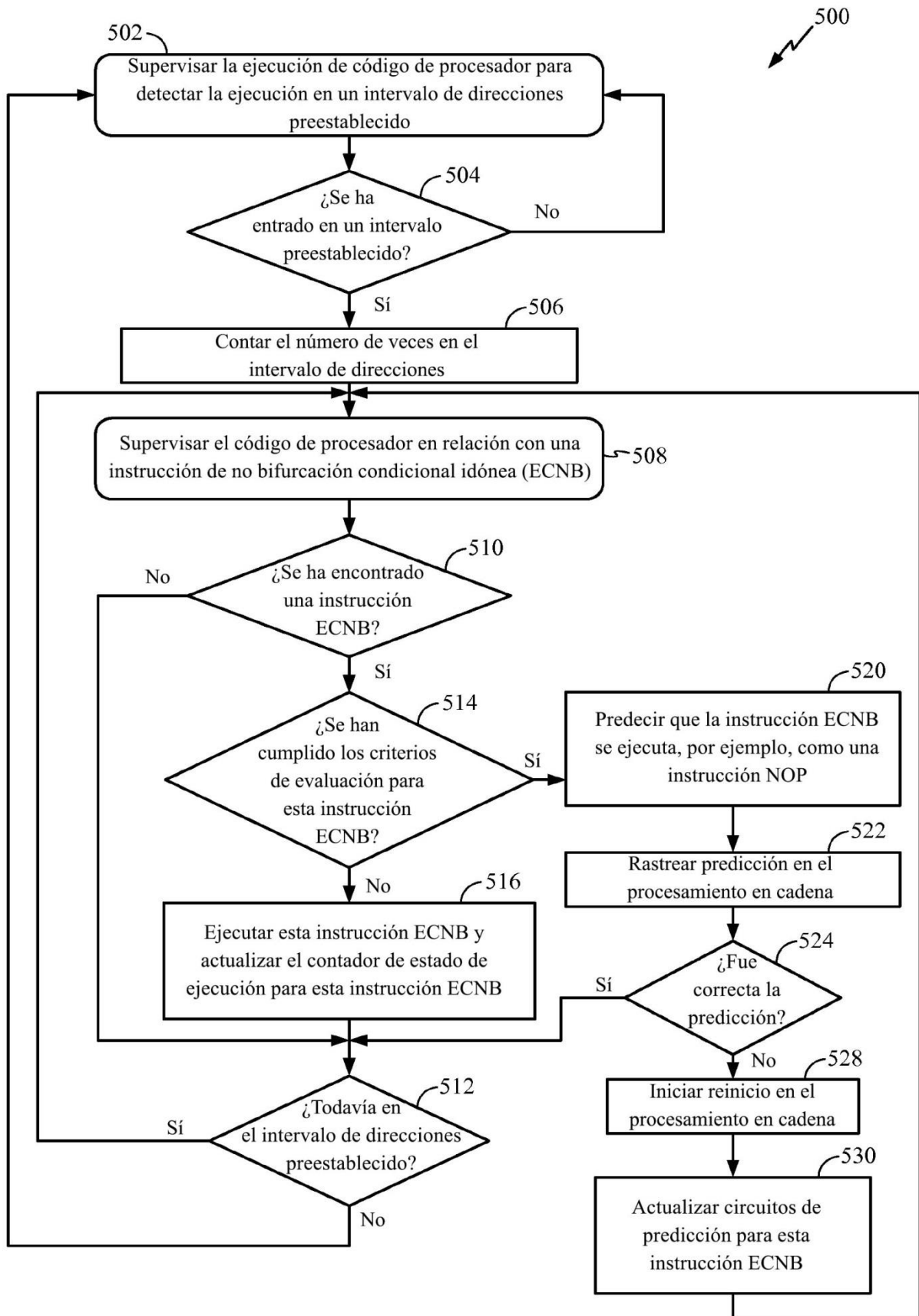


FIG. 5

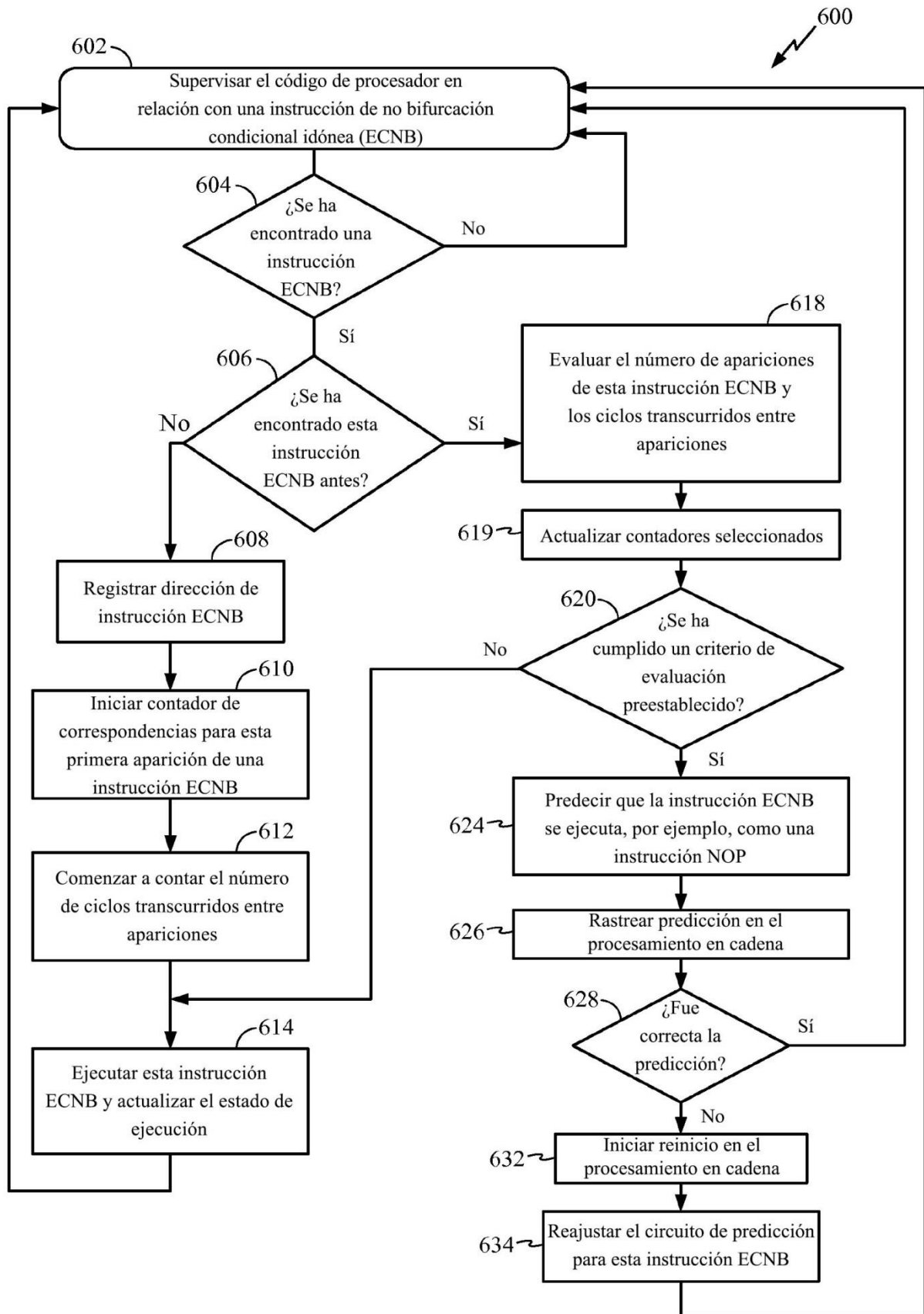


FIG. 6